F

SRI International

Annual Report, A010 • March 7, 1994

The NIDES Statistical Component Description and Justification

Harold S. Javitz
Statistics Program

Alfonso Valdes
Applied Electromagnetics and Optics Laboratory

SRI Project 3131
Contract N00039-92-C-0015

Prepared for:

Department of the Navy
Space and Naval Warfare Systems Command
2451 Crystal Drive
Arlington, VA 22245-5200
Attn: SPAWAR 02-22B LCDR Greg Breen
Attn: SPAWAR OOIE2, Robert D. Patton
Attn: SPAWAR PD51E

NRaD, Code 412
NRL, Code 5540
NSA, R23

Preface

SRI International has prepared this document as a full and complete disclosure of the Next-Generation Intrusion-Detection Expert System (NIDES) [3] statistical algorithm, including how it works, what decisions influenced the form of the algorithm, and the rationale behind those decisions. We have divided this document into four sections:

- Chapter 1 is a description of the NIDES statistical algorithm. It describes what the algorithm is and how it functions.
- Chapter 2 is a broad justification for the NIDES statistical algorithm. This section also includes a comparison to other statistical approaches to intrusion detection.
- Chapter 3 is a set of statistical criteria that can be used to evaluate the appropriateness of any statistical approach to intrusion detection. Although we did not formally use these criteria in the development of the NIDES statistical algorithm, they nevertheless had an important influence on the development of the algorithm. They may also be used to evaluate the suggestions of other statistical algorithm developers.
- Chapter 4 is a set of specific questions and answers that can be posed about the NIDES statistical algorithm. In this section we explore in more depth the specific choices we made in developing the NIDES statistical algorithm. We have found it convenient to use the question-and-answer format to address the relationship of the NIDES statistical algorithm to the work of Helman et al.

Contents

Chapter 1

Description of the NIDES Statistical Component

1.1 Overview of Statistical Component

The SRI NIDES statistical component observes behavior on a monitored computer system and adaptively learns what is normal for individual subjects: users, groups, remote hosts and the overall system. Observed behavior is flagged as a potential intrusion if it deviates significantly from expected behavior. The NIDES statistical component maintains a statistical subject knowledge base consisting of profiles. A profile is a description of a subject's normal (i.e., expected) behavior with respect to a set of intrusion-detection measures. Profiles are designed to require a minimum amount of storage for historical data and yet record sufficient information that can readily be decoded and interpreted during anomaly detection. Rather than storing all historical audit data, the profiles keep only statistics such as frequencies, means, and covariances.

The deductive process used by NIDES in determining whether behavior is anomalous is based on statistics, controlled by dynamically adjustable parameters, many of which are specific to each subject. Audited activity is described by a vector of intrusion-detection measures (or variables). Measures can be turned "on" or "off" (i.e., included in the statistical tests), depending on whether they are deemed to be useful for the monitored system. As each audit record arrives, the relevant profiles are retrieved from the knowledge base and compared with the vector of intrusion-detection measures. If the point in N-space defined by the vector of intrusion-detection measures is sufficiently far from the point defined by the expected values stored in the profiles, then the record is considered anomalous. Thus, NIDES evaluates the total usage pattern, not just how the subject behaves with respect to each measure considered singly.

The statistical knowledge base is updated daily, using the most recent day's observed behavior of the subjects. Before the new audit data are incorporated into

the profiles, the frequency tables in each profile are aged by multiplying them by an exponential decay factor. Although this factor can be set by the security officer, we believe that a value that reduces the contribution of knowledge by a factor of 2 for every 30 days is appropriate (this is the long-term profile half-life). This method of aging has the effect of creating a moving time window for the profile data, so that the expected behavior is influenced most strongly by the most recently observed behavior. Thus, NIDES adaptively learns subjects' behavior patterns; as subjects alter their behavior, their corresponding profiles change.

1.2 The NIDES Score Value

For each audit record generated by a user, NIDES generates a single test statistic value (the NIDES score value, denoted T^2) that summarizes the degree of abnormality in the user's behavior in the near past. (The concept of near past is defined later.) Consequently, if the user generates 1000 audit records in a day, there will be 1000 assessments of the abnormality of the user's behavior. Because each assessment is based on the user's behavior in the near past, these assessments are not independent.

Large values for T^2 are indicative of abnormal behavior, and values close to zero are indicative of normal behavior (e.g., behavior consistent with previously observed behavior). Thus, the security officer should be more concerned about larger values of T^2 than with smaller values.

Using historical information about T^2, cutoff values can be calculated corresponding to various alert levels (with higher alert levels associated with higher T^2 values). Each alert level is associated with a corresponding false positive rate (the probability that a normal user's activities will falsely be declared to be anomalous). The false positive rate can be expressed in two general ways: (1) as the proportion of audit records generated by the normal user that will exceed the threshold for declaring an audit record to be anomalous, or (2) as the probability that a normal user will be declared anomalous sometime during an average day. Currently, we have implemented only the first definition. In future versions of NIDES, we may implement the second definition. When multiplied by the number of system users, the latter definition corresponds loosely to the amount of effort that will be expended by the security officer in tracking down false leads. We have expressed the true positive rate (i.e., the probability that abnormal activity will be declared to be anomalous) as the proportion of "guest" user audit records that exceed the detection threshold (i.e., the proportion of one user's normal audit records that are declared anomalous when "played" through another user's profile, as if the first user had logged on as an uninvited "guest" in the second user's account). If a suite of intrusion scenarios is developed, the definition of true positive rates can be changed to the percentage of the intrusion scenarios that are detected. The security officer decides what actions should correspond to various alert levels, based partially upon the number of false leads that he or she can pursue and partially upon the system security needs. For example, the security officer might

choose to completely ignore the lowest alert levels, and be notified only at alert levels corresponding to an average of three false alerts per day.

Because the T^2 statistic summarizes behavior over the near past, and sequential values of T^2 are dependent, the T^2 values will slowly trend upward or downward.

To avoid inundating the security officer with notification of continued alerts we notify the security officer only when a change occurs in the alert level. We also "clear" the alert whenever the T^2 value becomes sufficiently low[1].

1.3 How T^2 is Formed from Individual Measures

The T^2 statistic is itself a summary judgment of the abnormality of many measures taken in aggregate. Suppose that there are n such constituent measures, and let us denote these individual measures by S_i, $1 \leq i \leq n$. Each S_i is a measure of the degree of abnormality of behavior with regard to a specific feature (such as CPU usage or file accesses). In the current version of NIDES, the T^2 statistic has been set equal to the sum of the squares of the S_i:

$$T^2 = ({S_1}^2 + {S_2}^2 + \cdots + {S_n}^2)/n$$

Because the T^2 statistic is a simple average of the n squares of the S_i, T^2 does not explicitly address the correlations among the S_i. We believe that there is additional useful information contained in the correlations among the S_i. Subsequent versions of NIDES could explore ways of introducing this covariation by defining a statistic L^2 as follows:

$$L^2 = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j>1} h(S_i, S_j, C_{ij})$$

Here, $h(S_i, S_j, C_{ij})$ is a well-behaved function of S_i, S_j, and their correlation C_{ij} that takes large values when S_i and S_j are not behaving in accordance with their historical correlations. An audit record would be declared to be abnormal when either T^2 or L^2 exceeded an appropriate threshold.

1.4 Types of Individual Measures

The individual S measures each represent some aspect of behavior. For example, an S measure might represent file accesses, CPU time used, or terminals used to log on. Two S measures might also represent only slightly different ways of examining

[1]The false positive and true positive rates should be calculated as if the modification of the alerting mechanism (to avoid inundating the security officer) had not been implemented. For example, the true positive rate is the proportion of intruder audit records detected as exceeding the threshold, whether or not alerts after the first were suppressed.

the same aspect of behavior. For example, both S_i and S_j might represent slightly different ways of examining file access.

We have found it useful to classify the different types of individual measures in the NIDES statistical system into the following four classes:

- *Intensity measures* — These three measures track the number of audit records that occur in different time intervals, on the order of 1 minute to 1 hour. These measures can detect bursts of activity or prolonged activity that is abnormal, primarily based on the volume of audit data generated.

- *Audit record distribution measure* — This single measure tracks all the types of activity that has been generated in the recent past, with the last few hundred audit records having the most influence on the distribution of activity types. For example, we might find that the last 100 audit records contained 25 audit records indicating that files were accessed, 50 audit records indicating that CPU time was incremented, 30 audit records indicating that I/O activity occurred, and 10 audit records indicating activity from a remote host. These data are compared to a profile of previous activity (generated over the last few months) to determine whether or not the distribution of activity types generated in the recent past (i.e., the last few hundred audit records) is unusual.

- *Categorical measures* — These are transaction-specific measures for which the outcomes are categories. For example, categorical measures could include the names of files accessed, the ID of the terminals used for logon, and the names of the remote hosts used. For the categorical measure of the names of files accessed, the individual categories within that measure are the file names themselves. The names of the files that were used in the last 100 to 200 audit records containing file names are compared to a historical profile of file names used to determine if recent usage is abnormal.

- *Counting measures* — These are measures for which the outcomes are counts. For example, counting measures might include CPU time (which counts the number of seconds of CPU used, with accuracy to 0.001 second) or the amount of I/O. Behavior over the last 100 to 200 audit records is compared to a historical profile of behavior to determine if recent usage is abnormal.

These different classes of measures serve different hierarchical purposes. The intensity measures assess the extent to which the volume of audit records generated is normal. The audit record distribution measure assesses, over the last few hundred audit records generated, the extent to which the types of measures being affected are normal. The categorical and counting measures assess within a type of audit record (e.g., an audit record that involves accessing a file, or that involves incrementing CPU time), the extent to which the behavior is normal over the past few hundred audit records.

1.5 Algorithm for Computing S from Q for the Intensity Measures

Each S measure is derived from a corresponding statistic that we will call Q. In fact, each S measure is a "normalizing" transformation of the Q statistic so that the degree of abnormality for different types of features (such as CPU usage and the names of files accessed) can be added on a comparable basis. Two different methods for transforming the Q statistics into S values are used. One method is used for computing the values of S corresponding to the three intensity measures; a second method is used for computing the values of S corresponding to all the other measures.

For the intensity measures, the value of Q corresponding to the current audit record represents the number of audit records that have arrived in the recent past. Here, "recent" past corresponds to the last few minutes for the Q statistic with a half-life of 1 minute and to the last few hours for the Q statistic with a half-life of 1 hour. In addition to knowing the current value for Q, NIDES maintains a historical profile of all previous values for Q. Thus, the current value of Q can be compared to this historical profile to determine whether the current value is anomalous.

The transformation of Q to S for the intensity measures requires knowledge of the historical distribution of Q. For example, we might find the following historical information for the intensity measures Q with a half-life of 1 minute:

- 1% of the Q values are in the interval 0 to 10 audit records
- 7% are in the interval 10 to 20
- 35% are in the interval 20 to 40
- 18% are in the interval 40 to 80
- 28% are in the interval 80 to 160
- 11% are in the interval 160 to 320

The S statistic would be a large positive value whenever the Q statistic was in the interval 0 to 10 (because this is a relatively unusual value for Q) or whenever Q was larger than 320 (because this value has not historically occurred). The S statistic would be close to zero whenever Q was in the interval 20 to 40, because these are relatively frequently seen values for Q. The selection of appropriate intervals for categorizing Q is important to the functioning of the algorithm. We are currently using 32 intervals for each Q measure, with interval spacing being either linear or geometric. The last interval does not have an upper bound, so that all values of Q belong to some interval.

Small values of Q are indicative of a recent past that is similar to historical behavior, and large values of Q are indicative of a recent past that is not similar to

historical behavior. This induces a modification in the transformation of Q to S, so that S is small whenever Q is small, and S is large whenever Q is large. Hence, S can be viewed as a type of rescaling of the magnitude of Q.

The algorithm for converting individual Q values to S values for the intensity measures (but not for other measures) is as follows:

1. Let P_m denote the relative frequency with which Q belongs to the m^{th} interval. In our example, the first interval is 0 to 10 and the corresponding P value (say P_0) equals 1%. There are 32 values for P_m, with $0 \leq m \leq 31$.

2. For the m^{th} interval, let $TPROB_m$ denote the sum of P_m and all other P values that are smaller than or equal to P_m in magnitude. In our previous example, $TPROB$ for the interval of $40 \leq Q \leq 80$ equal to 18% + 11% + 7% + 1% = 37%.

3. For the m^{th} interval, let s_m be the value such that the probability that a normally distributed variable with mean 0 and variance 1 is larger than s_m in absolute value equals $TPROB_m$. The value of s_m satisfies the equation

$$P(|N(0,1)| \geq s_m) = TPROB_m$$

or

$$s_m = \Phi^{-1}(1 - (TPROB_m/2))$$

where Φ is the cumulative distribution function of a N(0,1) variable. For example, if $TPROB_m$ is 5%, then we set s_m equal to 1.96, and if $TPROB_m$ is equal to 100%, then we set s_m equal to 0. We do not allow s_m to be larger than 4.0.

4. Suppose that after processing an audit record we find that the Q value is in the m^{th} interval. Then S is set equal to s_m, the s value corresponding to $TPROB_m$.

1.6 Algorithm for Computing S from Q for All Other Measures

For all measures other than the intensity measures, Q compares short-term behavior to long-term behavior. For example, for the command usage measure, Q measures the extent to which the most recent few hundred commands issued are consistent with long-term command usage.

For both the intensity measures and the other measures, we calculate a long-term profile for Q using 32 intervals. For example, for a non-intensity measure such as names of commands used we might find a probability distribution for Q similar to

the one displayed earlier for an intensity Q, except that the range of values into which Q could be classified would not be in units of audit records. Rather, the range of values would be expressed in terms of the degree of similarity between the short-term profile of command usage and the long-term profile of command usage, with larger numbers representing less similarity.

Because of the difference in the way that Q is defined for intensity measures and other measures, the transformation of Q to S is slightly different for non-intensity measures. For non-intensity measures, we let $TPROB_m = P_m + P_{m+1} + \cdots + P_{31}$. In our previous example, if Q were a non-intensity measure, the $TPROB$ value of the interval $40 \leq Q \leq 80$ would be equal to $18\% + 28\% + 11\% = 49\%$. Thus, in these cases, S is a simple mapping of the percentiles of the distribution of Q onto the percentiles of a half-normal distribution.

In practice these algorithms are easy to implement, with the Q tail probability calculations done only once — at update time (usually close to midnight). (The s_i values, requiring only a table look up, are done in real time.) Each interval for Q is associated with a single s value, and when Q is in that interval, S takes the corresponding s value.

1.7 Computing the Frequency Distribution for Q

The historical frequency distribution for Q is required for Q to be transformed into S. The calculation procedures described here are used for all types of measures.

The first step in calculating the historical probability distribution for Q is to define bins into which Q can be classified. We always use 32 bins (numbered 0 to 31) for a measure Q. Let Q_{max} be the maximum value that we ever expect to see for Q. This maximum value depends on the particular types of measures being considered. Default values are provided in NIDES for these maximum values and they should be reset by the security officer if Q is in the highest bin more than 1% of the time. The cut points for the 32 bins are defined on either a linear or geometric scale. For example, when a geometric scale is used, bin 0 extends from 0 to ${Q_{max}}^{1/32}$, bin 1 extends from ${Q_{max}}^{1/32}$ to ${Q_{max}}^{2/32}$, bin 2 extends from ${Q_{max}}^{2/32}$ to ${Q_{max}}^{3/32}$, and bin 31 extends from ${Q_{max}}^{31/32}$ to infinity.

As before, let P_m denote the relative frequency with which Q is in the m^{th} interval (i.e., bin). Each Q statistic is evaluated after each audit record is generated (whether or not the value of Q has changed), and therefore P_m is the percentage of all audit records for which Q is in the m^{th} interval.

The formula for calculating P_m on the k^{th} day after initiating NIDES monitoring of a user is:

$$P_{m,k} = (1/N_k) \sum_{j=1}^{k} (W_{m,j} 2^{-b(k-j)})$$

where

k = the number of days that have occurred since the user was first monitored

b = the decay rate for P_m that determines the half-life of the data used to estimate P_m; we currently recommend a 30-day half-life, corresponding to a b value of $-log_2(0.5)/30 = 0.0333$

$W_{m,j}$ = the number of audit records on the j^{th} day for which Q was in the m^{th} interval

N_k = the exponentially weighted total number of audit records that have occurred since the user was first monitored

The formula for N_k is:

$$N_k = \sum_{j=i}^{k} W_j 2^{-b(k-j)}$$

where

W_j = the number of audit records occurring on the j^{th} day

The formula for $P_{m,k}$ appears to involve keeping a long sum (e.g., since monitoring first began), but the computations are simplified by using the following recursion formulas:

$$P_{m,k} = (2^{-b} P_{m,k-1} N_{k-1} + W_{m,k})/N_k$$

$$N_k = 2^{-b} N_{k-1} + W_m$$

In NIDES, we update $P_{m,k}$ and N_k once per day and keep running totals for $W_{m,k}$ and W_k during the day.

1.8 Computing the Q Statistic for the Intensity Measures

When a user is first audited, that user has no history. Consequently, we must choose some convenient value to begin the Q statistic history. For example, we might initially let each Q measure be zero, or some value close to the mean value for other similar users.

Each Q statistic for intensities is updated each time a new audit record is generated. Let us now consider how to update Q. Let Q_n be the value for Q after the n^{th} audit record, and Q_{n+1} be the value for Q after the $(n+1)st$ audit record. The formula for updating Q is:

$$Q_{n+1} = 1 + 2^{-rt}Q_n$$

where

- The variable t represents the time (say in minutes or fractions thereof) that has elapsed between the n^{th} and $(n+1)st$ audit records.
- The decay rate r determines the half-life of the measure Q. Large values of r imply that the value of Q will be primarily influenced by the most recent audit records. Small values of the decay rate r imply that Q will be more heavily influenced by audit records in the more distant past. For example, a half-life of 10 minutes corresponds to an r value of $0.10 = log_2(0.5)/10.0$. The security officer may set the half-life of the intensity measures at any values that he or she feels are appropriate. We are currently using three intensity measures with half-lives of 1, 10, and 60 minutes, respectively.

Q is the sum of audit record activity over the entire past usage, exponentially weighted so that the more current usage has a greater impact on the sum. Q is more a measure of near past behavior than of distant past behavior, even though behavior in the distant past also has some influence on Q. The Q statistic has the important property that it is not necessary to keep extensive information about the past to update Q.

We note that the intensity measures use clock time as the unit by which age is calculated. This is important because the intent of this measure is to assess the extent to which bursts of activity are normal. All the other measures in NIDES determine "age" using audit record counts. For example, an audit record may be the most recent record (that affects that measure), the second most recent, the third most recent, and so forth. This assures profile continuity over nights and weekends — periods when there is typically little activity. Thus, for the non-intensity measures we address the issue of whether over the past few hundred audit records that affected the measure, behavior was normal in comparison with historical standards, regardless of when that activity took place. (Although at first glance this means that abnormal "short term" behavior might include behavior from days, weeks, or even months in the past, this is not a concern, since the portion of the activity in the short-term profile that has taken place earlier than the last profile updating will have already been incorporated into the historical profile, and will tend to be assessed as normal).

1.9 Computing the Q Statistic for the Audit Record Distribution Measure

Each audit record that is generated indicates one or more types of activity that have occurred for a user. For example, a single audit record may indicate that a file has

been accessed, that I/O has occurred, and that these activities occurred from a remote host. The Q statistic for the audit record distribution measure is used to evaluate the degree to which the type of activity in the recent past agree with the distribution in a longer-term profile.

The calculation of the audit record distribution measure begins with the specification of the types of activities that will be examined. We currently recommend that each categorical and counting measure (with some exceptions as noted below) constitute an activity type. For example, if name of file accessed is a categorical measure, then the corresponding activity would be that any named file was accessed. Similarly, if amount of I/O used is a counting measure, then the corresponding activity would be that any I/O was used. That is, if an audit record would cause a categorical or counting measure to be recalculated, then a corresponding activity type should be defined. The exception would be a categorical or counting measure that would be affected by any audit record. For example, if hour of audit record generation is a categorical measure, then every audit record causes this categorical measure to be updated and no purpose is served in defining a corresponding activity type. We note that in general a single audit record can impact multiple measures. (That is, a single audit record can "touch" multiple measures.)

It may be useful to define activity types in addition to those based on the categorical and counting measures used. For example, the security officer may want to evaluate the percentage of audit records generated that indicate that the user is logged onto a remote host. The security officer who is not interested in which remote host is being used may accomplish this goal in one of two ways. The first way is to establish a categorical measure with two categories — "remote host indicated" and "remote host not indicated". If this approach is used, then there should be no corresponding activity type defined because every audit record is relevant to the calculation of the categorical measure. The second way is to define an activity type of "remote host indicated" to be used by the audit record distribution measure and not to define a categorical measure. These two approaches yield similar (although not totally equivalent) results, but the second approach is computationally less intensive.

Suppose that we have established M activity types. For each activity we must calculate a long-term historical relative frequency of occurrence, denoted f_m, for that activity type. For example, suppose that over the last 3 months, 7% of all audit records have involved file accesses. Then f_m for the file access activity type would be 0.07. Note that each f_m is between 0 and 1.0 inclusive. The sum of the f_m may be greater than 1.0 because a single audit record may indicate that multiple activity types have occurred.

The algorithm used to compute f_m is essentially the same as that used to calculate P_m and uses the same decay rate. That is, we may write that the value of f_m on the k^{th} day is equal to

$$f_{m,k} = (1/N_k) \sum_{j=1}^{k} (W_{m,j} 2^{-b(k-j)})$$

where N_k, and b are defined as before and $W_{m,j}$ is the number of audit records on the j^{th} day that indicate that the m^{th} activity type has occurred.

In NIDES, the Q statistic compares the short-term distribution of the types of audit records that have been generated with the long-term distribution of types. In the simplest situation, Q_n (the value of the Q statistic when the n^{th} audit record is processed) is defined as follows:

$$Q_n = \sum_{m=1}^{M} [(g_{m,n} - f_m)^2 / V_m]$$

where

$g_{m,n}$ = the relative frequency with which the m^{th} activity type has occurred in the recent past (which ends at the n^{th} audit record)
V_m = the approximate variance of the $g_{m,n}$

If we view $g_{m,n}$ as the short-term profile for the audit record distribution and we view f_m as the long-term profile for audit record distribution, then Qn measures the degree of dissimilarity between the short-term behavior and long term-behavior. That is, Q_n is larger whenever the distribution of activity types in the recent past differs substantially from the historical distribution of activity types, where "substantially" is measured in terms of the statistical variability introduced because the near past contains relatively small (effective) sample size. The value of $g_{m,n}$ is given by the formula

$$g_{m,n} = (1/N_r) \sum_{j=1}^{n} [I(j,m) 2^{-r(n-j)}]$$

or by the recursion formula

$$g_{m,n} = 2^{-r} g_{m,n-1} + [I(n,m)/N_r]$$

where

j = an index denoting audit record sequence
$I(j,m) = 1.0$ if the j^{th} audit record indicates activity type m has occurred and 0.0 otherwise

r = the decay rate for Q that determines the half-life for the Q measure; we have set the half-life to approximately 100 audit records, corresponding to an r value of $-log = (0.5)/100 = 0.01$.

N_r = the sample size for the Q statistic, which is given by the formula

$$N_r = \sum_{j=1}^{n} 2^{-r(n-j)}$$

and which rapidly approaches an asymptotic value of $1/(1-2^{-r})$

The value of V_m is given by the formula

$$V_m = f_m(1-f_m)/N_r$$

except that V_m is not allowed to be smaller than $0.01/N_r$.

1.10 Computing the Q Statistic for Categorical Measures

Categorical measures are those that involve the names of particular resources being used (such as the names of files being accessed, or the location from which logons are attempted) or involve other categorical characteristics of audit records, such as the hour of the day on which the audit record was generated.

The method used for computing Q for categorical measures is essentially the same as that previously described for computing Q for the audit record distribution measure. In fact, we may view the audit record distribution measure as a categorical measure. The only difference in the definition of Q is that every audit record results in the recalculation of the audit record distribution measure, whereas for all other categorical measures, Q is only updated whenever the audit record contains information relevant to the particular measure. For example, the file name accessed measure would only be updated whenever the audit record concerns a file access. A half-life for the file name accessed measure of 100 audit records would refer to 100 audit records relevant to file names accessed, rather than to the last 100 audit records.

1.11 Computing the Q Statistic for Counting Measures

Counting measures are those that involve counts of particular resources used (such as CPU time in milliseconds or I/O counts) or some other numerical feature of audit records (such as the interarrival time in milliseconds of consecutive audit records).

Counting measures are transformed to categorical measures. For example, consider the counting measure of CPU time. Individual audit records arrive that indicate that a non-zero amount of CPU time has been incrementally expended since the last reporting of CPU usage. We might expect that this delta-CPU value would be between 0 and a maximum of 200 seconds. We define 32 geometrically scaled intervals between 0 and 200 milliseconds employing the same procedure as we used for defining intervals for the historical profile for Q (including the convention that the last interval actually extends to infinity). When a delta-CPU value arrives and is classified into interval m, we state that a categorical event m has occurred. Thus, we translate CPU time into a categorical measure where a category is activated whenever an audit record arrives with a delta-CPU value in that category. Thus, the Q measure for CPU time doesn't directly measure total CPU usage in the near past; rather, it measures whether the distribution of delta-CPU values in the near past is similar to the historical distribution of delta-CPU values. Once the counting measure is redefined as a categorical measure as discussed above, the Q statistic is calculated in the same fashion as for any other categorical measure.

1.12 Addition of a "Rare" Category

In previous versions of NIDES, each category of behavior in a particular measure was treated separately when comparing the long- and short-term profiles. For example, for the measure of "files accessed," each different file name in the long-term profile was treated as an individual category. Often, this resulted in hundreds of categories in the long-term profile, the vast majority of which had very small probabilities (much less than 1%).

The presence of so many categories makes it difficult to detect intruders. An intruder browsing the host's files would tend to touch a reasonable number of the files that the user doesn't use very often. Thus, the intruder might demonstrate a short-term profile with small but non-negligible probabilities (say on the order of 2%) for many of these files. On a file-by-file basis, a comparison of two small probabilities would not be statistically significant.

We decided that if we aggregated the rarest categories, then the detection of an intruder who frequently touched rare categories would be easier. We now create a temporary new "rare" category (at profile updating time) that contains all of the user's files with very small probabilities. Categories are temporarily added to the rare category until the addition of another category would cause the cumulative sum of the probabilities that have already been added to the rare category to first exceed a preset threshold for the sum of rare probabilities (for example, 1%). When the short-term profile is compared to the long-term profile, all of the files that are combined into the new rare category are treated as if they were the same file. Thus, an intruder might have a short-term profile in which 30% of the probability was in the rare category. The difference between the actual rare category probability sum in the long-term

profile and 30% probability in the short-term profile will tend to be easier to detect than the differences between very small probabilities.

We note that the use of the rare category is not without trade-offs. For example, if the intruder continues to use a single file in the rare category, this will be more difficult to detect than previously. We also note that separate counts for all categories are maintained (whether or not a particular category has temporarily become part of the rare category for a particular day); this allows categories to migrate into and out of the rare category on different days.

1.13 Addition of a "New" Category

One type of behavior that we need to be particularly careful to identify is the creation of, or use of, new categories. The most obvious example would be the use of commands that the host user has never used before. (Because our profiles eventually discard categories with associated probabilities that decay below certain limits, it is more precise to say that these are categories that have not been used recently by the host.)

To increase our ability to detect new behavior within a measure, we have added a "new" category. The first time during a day when a short-term behavior is noticed for which there is no category in the long-term profile, the short-term probability of the new category is incremented. This is compared to the probability in the long-term profile for the new category to determine whether the amount of new behavior is statistically significant.

For example, suppose that the probability in the long-term profile for the new category is 0.1%. (This will almost always be a very small probability.) Further suppose that there are no categories in the long-term profile for commands W, X, Y, or Z. Suppose that the short-term profile is effectively based on 200 audit records, and that until now no new short-term behavior has been observed over, say, the last 1000 audit records.) Finally, suppose that 10 new audit records arrive, including two occurrences of X, three of Y, one of Z, and one of W. The short-term probability of the new category would be $4/200 = 2\%$[2]. In the calculation of Q we include a term for the "new category" with an observed percentage of 2% and an expected percentage of 0.1% (e.g., an observed count of about 4 and an expected count of about 0.2).

[2]The probability for the new category is calculated separately from the probabilities for all other categories. Thus, if there is a measure with many new categories (such as a file access measure) the probabilities for all other categories still sum to 100%.

Chapter 2

Rationale for the Current NIDES Statistical Component

The objective of describing the rationale for the NIDES statistical component is fourfold:

- To justify the statistical algorithms that SRI is currently using in NIDES
- To discuss how the NIDES approach relates to those used in two other intrusion-detection systems (Wisdom and Sense and Haystack)
- To discuss how the NIDES approach relates to four general statistical approaches (pattern recognition, discriminant and classification analysis, Marcovian transition analysis, and Bayesian decision analysis)

2.1 The Current NIDES Statistical Component

2.1.1 Statistical Component Philosophy

Understanding why SRI has chosen to implement certain types of algorithms in the NIDES statistical component requires an explanation of the philosophy underlying the statistical component. A critical element in the development of this philosophy was the knowledge that the NIDES statistical component would be used in conjunction with a rule-based component. It was assumed that the rule-based component would be defined to detect all known methods of compromising system security (or detect indicators that system security is or might be compromised). The statistical component is therefore devoted to uncovering all anomalous behaviors, just in case these anomalous behaviors might imply that system security is being compromised via some unknown or otherwise undetectable method. It may be the only way of detecting an insider attack, which may be composed of actions that in other circumstances would be considered acceptable or which exploit previously unknown system vulnerabilities.

The basic philosophy underlying the statistical component is that it is intended as the detection or protection system of last resort. It is intended to uncover those actions that cannot be prevented (e.g., through physical or software safeguards, such as passwords) or detected by a set of rules embedded in a rule-based component. Thus, whenever we (i.e., the intrusion-detection community) know that a particular set of actions either constitutes a violation of system security or is sufficiently serious to warrant an investigation or immediate preventive steps (such as disconnecting the user), the statistical component assumes that such set of actions is either prevented via physical or software safeguards or encoded as a set of rules in a rule-based component.

Because it is intended as a detection system of last resort, the statistical component cannot rely on our previous knowledge of security violations or methods of compromising security. It must assume that the user might be engaging in a totally new or previously unknown method of compromising security. It cannot therefore rely on the body of knowledge that we have previously gathered. All that previous knowledge is assumed to be embedded in the rule-based component. (However, testing the statistical component against known intrusion, masquerading, or malfeasance scenarios is valuable in that it provides us with some level of assurance that the statistical component is capable of detecting these types of anomalous behavior.)

Because it cannot rely on previous knowledge of methods to compromise security, the statistical component attempts to detect any anomalous behavior, whether or not that behavior has been previously associated with malfeasance. Thus, if the user engages in unusual behavior (where the unusualness of behavior is measured relative to that user's own prior behavior or relative to the behavior of a group of employees who are supposed to be using the computer system in the same way), the statistical component raises a warning flag. The warning flag does not mean that system security is being compromised — only that the user is behaving in an unusual fashion. For example, if a user suddenly starts accessing directories that he or she has not previously accessed, this anomaly would be detected even if the user has full and legitimate access to those directories. On the other hand, if a user *routinely* exploits a flaw in the operating system to change his or her access privileges to "root" and then back again, then this behavior would not be deemed anomalous even though this action violates system security.

2.1.2 Basic NIDES Statistical Approach

The basic statistical approach in NIDES is to compare a user's short-term behavior to the user's long-term behavior. Whenever short-term behavior is sufficiently unlike long-term behavior, a warning flag is raised. As discussed above, this statistical approach requires no *a priori* knowledge about what types of behaviors would lead to compromised security.

The number of audit records or number of days that constitute short-term and long-term behavior can be set by the security officer through the specification of a

"half-life." The NIDES developers will provide "rules of thumb" for the specification of the half-life. For example, a security officer who wants short-term behavior to be on the order of 200 audit records should specify a half-life of approximately 100 audit records. This will assure that the 200th audit record has only one-quarter the influence of the most recent audit record, the 400th audit record has only one-sixteenth the influence, and so forth. Similarly, a reasonable half-life for a long-term profile might be 30 days.

In comparing short-term behavior to long-term behavior, the statistical component will be concerned about long-term behaviors that do not appear in short-term behavior as well as about short-term behaviors that are not typical of long-term behavior. For example, if a particular file is the object of 10% of file accesses in the long term, whereas in the current short-term behavior only 1% of file accesses involve that file, then this discrepancy will contribute towards a finding of abnormality. Similarly, if a particular file is the object of only 1% of file accesses in the long term, whereas in the current short term this file is the object of 10% of file accesses, then this discrepancy will contribute towards a finding of abnormality.

2.1.3 Time Horizon of Short-Term Behavior in NIDES

The NIDES statistical component keeps track of many different measures (or aspects) of behavior, such as commands used, files accessed, and remote hosts accessed. For each of these measures, it keeps track of short-term behavior separately. Therefore the half-life of short-term behavior for commands used refers to the last (say) 100 commands issued, whether these commands were executed in the last minute, hour, day, or week. The half-life of short-term behavior for files accessed refers to the last (say) 100 files accessed, whether these files were accessed in the last minute, hour, day, or week. Thus, the chronological time frames for the different measures can be different. In addition, it is not theoretically necessary that short term behavior for each measure refer to the same half-life (although the current implementation of NIDES does not allow easy specification of measure-specific half-lives). For example, the half-life of short-term behavior with respect to remote host logins could be 20 logins, whereas the half-life of short-term behavior with respect to file accesses might refer to the last 500 file accesses. We hope in the future to provide automated support to setting half-lives, so that the number of audit records that constitute the half-life of short-term behavior for each measure is large enough to ensure stability in the measure (making it easier to achieve a low false positive warning rate) and small enough that the measure will react rapidly to an intrusion.